

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA**



Organización y Arquitectura de Computadoras

TAREA COMPLEMENTARIA DE LABORATORIO

Alumno:

Daniel Ernesto Rivas Pareja (20212475)

Docente:

Benavides Aspiazu, Jorge

Lima, Perú

2023-2

Respuestas

1) Parte D

Primero, como la versión síncrona del servidor, las operaciones de entrada/salida se dan de manera secuencias, es decir, una detrás de otra. Por esto último, solo se puede manejar una solicitud del cliente a la vez debido a que durante el tiempo de espera el hilo principal del programa se bloquea y nos impide manejar otras solicitudes entrantes. Por lo tanto, esto nos crea una limitación en cuanto a la capacidad del servidor para manejar múltiples conexiones concurrentes de manera eficiente.

Por otro lado, la versión asíncrona de nuestro servidor al ser asíncrono nos permite que el servidor maneje múltiples conexiones entrantes de mejor manera y sin bloquear el hilo principal del programa. Es decir, las operaciones de entrada y salida se ejecutan de manera concurrente lo que le permite al servidor manejar múltiples tareas al mismo tiempo. Por lo tanto, esto involucra una mejora significativa en cuanto a la capacidad de respuesta y rendimiento en general del servidor para grandes cantidades de solicitudes.

Analizando los tiempos obtenidos, obtuvimos que al hacer una misma solicitud del cliente a ambos servidores el tiempo de ejecución de la solicitud era considerablemente menor en el caso del servidor síncrono. Esto se debe a que el servidor de versión asíncrona es más eficiente para casos de un gran número de solicitudes ya que naturalmente este tipo de servidor está diseñado para tratar múltiples conexiones concurrentes y operaciones de entrada y salida. Por lo cual, para casos con pocas solicitudes puede pasar que el tiempo de ejecución de la solicitud sea mayor en la versión asíncrona debido a la carga adicional de asociada a la programación asíncrona como librerías, funciones y administración de tareas. Sin embargo, generalmente esta carga está justificada por los casos de alto volumen de solicitudes. De igual forma, es importante analizar los casos para los cuales se desarrolla estos servidores ya que si la versión síncrona es eficiente y más sencilla de desarrollar y además se tiene en cuenta que las solicitudes no lleguen a una gran cantidad entonces se podría considerar realizar una versión síncrona en lugar asíncrona.

2) Parte E

Como se mencionó anteriormente, el esquema asíncrono para nuestro servidor puede correr concurrentemente varias solicitudes de los clientes debido a que permite manejar múltiples conexiones entrantes sin dejar esperando a que termine las solicitudes previas para que se inicie la nueva solicitud. Esto debido a que como se dijo anteriormente el esquema asíncrono no bloquea el hilo principal del programa a comparación del esquema síncrono permitiendo este manejo concurrente de conexiones antes mencionado. Por otro lado, el esquema síncrono no puede realizar este manejo concurrente debido a su procesamiento secuencial para las solicitudes, por lo cual, esta versión si deja en espera a la nueva solicitud hasta que se complete la solicitud previa. Además, esto se prueba en el programa “prueba_solicitudes.py” donde se ve cómo se van enviando las solicitudes sin esperar a que termine anterior.

Cliente – Servidor Asíncrono - Tiempos (Parte D)

Nota recibida, cerrando conexion

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA\TAREA_20212475> python .\cliente.py
```

Ingresa el codigo del alumno: 20230010

Tiempo de transaccion : 1.136 ms

Nota final del alumno 20230010 es 11

Nota de alumno 20230013: 10

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA\TAREA_20212475> python .\cliente.py
```

Ingresa el codigo del alumno: 20230010

Tiempo de transaccion : 1.153 ms

Nota final del alumno 20230010 es 11

Nota recibida, cerrando conexion

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA\TAREA_20212475> python .\cliente.py
```

Ingresa el codigo del alumno: 20230010

Tiempo de transaccion : 1.091 ms

Nota final del alumno 20230010 es 11

Cliente – Servidor Síncrono - Tiempos (Parte D)

Nota recibida, cerrando conexion

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA\TAREA_20212475> python .\cliente.py
```

Ingresa el codigo del alumno: 20230010

Tiempo de transaccion : 0.612 ms

Nota final del alumno 20230010 es 11

Nota recibida, cerrando conexion

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA\TAREA_20212475> python .\cliente.py
```

Ingresa el codigo del alumno: 20230011

Tiempo de transaccion : 0.553 ms

Nota final del alumno 20230011 es 11

Nota recibida, cerrando conexion

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA\TAREA_20212475> python .\cliente.py
```

Ingresa el codigo del alumno: 20230185

Tiempo de transaccion : 0.550 ms

Nota final del alumno 20230185 es 10

Nota recibida, cerrando conexion

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA\TAREA_20212475> python .\cliente.py
```

Prueba Clientes – Servidor Asíncrono (Parte E)

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA> python .\prueba_solicitudes.py
Enviando solicitud de alumno 20230010
Enviando solicitud de alumno 20230011
Enviando solicitud de alumno 20230012
Enviando solicitud de alumno 20230013
Nota de alumno 20230010: 12
Nota de alumno 20230011: 10
Nota de alumno 20230012: 9
Nota de alumno 20230013: 8
Tiempo promedio por solicitud : 0.238 ms
```

Servidor Asíncrono (Parte E)

```
PS C:\Users\USER\Documents\Universidad\2023-2\OAC\TAREA> python .\servidor_asyn.py
Servidor de transferencia de notas escuchando en 127.0.0.1:5000
Conexion entrante desde ('127.0.0.1', 50081)
Tiempo de ejecucion de solicitud 3.048 ms
Archivo enviado, cerrando conexion
Conexion entrante desde ('127.0.0.1', 50082)
Tiempo de ejecucion de solicitud 0.534 ms
Archivo enviado, cerrando conexion
Conexion entrante desde ('127.0.0.1', 50083)
Tiempo de ejecucion de solicitud 0.566 ms
Archivo enviado, cerrando conexion
Conexion entrante desde ('127.0.0.1', 50084)
Tiempo de ejecucion de solicitud 0.577 ms
Archivo enviado, cerrando conexion
```